The Hong Kong Polytechnic University
Department of Computing
COMP4913 Capstone Project
RF-based indoor localization system

Presented by CHEUK Lok Kan (19053031D)

BSc (Hons) in Information Technology (61431-FIT)

Supervisor: Dr. YANG Ray

Background

- A research-intensive topic: Indoor radio frequency (RF) localization
- Inaccuracies due to errors of indoor localization
  - Hardware and spatial diversities
  - Multipath effects due to obstacles
  - Mathematical errors by coordinate conversion
- Scenario: Finding a particular book out of dozens of books placed on a large shelf in a library
- Accuracy is the priority

Why RFID?

- It owns the “Ray” database, an enormous, general-purpose, and protocol-free database consisting of over 1.32 million datasets collected in 14 scenes and 37 settings (ACM, 2022).

Env. (#)	Scene (#)	Setting (#)	RSS (dBm)	1G (#)	2G (#)	3G (#)	Total (#)	Density (p/m^3)	Space (m^2)	MP. (#)	Dist. (m)	Size (GB)	DR. (r/s)	TS. (min)	Temp. (°C)
Semi-Indoor Env.	A	S1	−62.5	32,191	35,308	16,893	84,392	3,843.0	78.5	10	5	48.24	40.4	79.5	31.2
		S2	−66.4	21,551	24,480	11,300	57,311	4,689.9	314.2	8	10	28.47	38.0	44.4	30.3
		S3	−66.7	20,372	23,773	11,382	55,527	5,274.2	706.9	7	15	26.88	42.4	41.2	29.9
		S4	−69.4	19,483	24,858	10,177	54,518	3,787.0	1,256.6	9	20	24.18	41.0	35.1	29.4
		S5	−71.0	16,567	22,278	11,457	50,302	4,336.4	1,963.5	10	25	25.38	40.1	35.0	27.2
		S6	−75.0	18,414	21,818	11,009	51,241	5,865.5	2,827.4	9	30	22.68	38.2	32.9	27.4
		S7	−77.4	16,445	22,807	12,037	51,289	5,871.0	3,848.5	10	35	23.82	35.9	33.8	27.7
		S8	−78.8	28,672	35,540	10,309	74,521	7,834.4	5,026.5	10	40	35.28	40.5	50.8	28.1
		S9	−79.3	28,672	35,540	10,309	61,909	4,235.7	6,361.7	8	45	27.66	41.4	40.9	28.3
		S10	−79.1	32,919	33,753	9,803	76,475	5,223.7	7,854.0	13	50	25.80	41.2	63.6	29.0
		S11	−88.6	29,440	17,387	3,359	50,186	10,490.4	7,854.0	11	55	18.15	31.5	29.8	28.7
	B	S12	−71.8	4,301	10,683	8,044	23,028	538.7	1,963.5	10	25	43.62	6.2	69.5	30.1
		S13	−76.9	6,245	10,172	4,940	21,357	702.1	3,848.5	13	35	39.07	5.9	66.9	30.4
		S14	−78.1	6,942	19,579	11,782	38,303	1,381.8	5,026.5	12	40	64.44	6.8	113.1	30.9
	C	S15	−68.3	5,533	9,075	4,118	18,726	6,079.9	1,256.6	16	20	40.02	7	74.5	33.1
	D	S16	−68.9	24,007	32,448	21,083	77,538	4,345.1	530.9	7	13	21.48	33.9	39.5	29.2
		S17	−67.0	15,684	16,584	8,303	40,571	2,545.9	530.9	10	13	46.92	32.2	81.6	28.8
Full-Indoor Evn.	E	S18	−66.2	2,326	31,693	126,475	160,494	38,212.9	314.2	14	10	89.93	20.7	61.2	18.4
		S19	−65.3	8,720	69,915	N/A	78,635	27,924.4	314.2	11	10	31.25	27.4	124.1	24.9
		S20	−63.7	3,173	26,930	N/A	30,103	10,906.9	314.2	11	10	12.19	16.1	26.5	25.1
		S21	−64.9	2,998	23,918	N/A	26,916	8,900.8	314.2	10	10	11.74	39.3	21.7	27.6
		S22	−65.4	18,872	13,170	N/A	32,042	5,057.1	314.2	11	10	12.47	40.7	18.0	24.8
	F	S23	−65.1	4,930	17,714	25,823	48,467	22,627.0	153.9	9	7	71.46	8.1	124.1	25.8
	G	S24	−61.4	1,749	4,222	4,550	10,521	4,911.8	78.5	8	5	11.88	4.2	65.9	27.5
		S25	−60.2	891	2,425	1,975	5,291	937.8	78.5	5	5	9.42	4.3	33.6	30.1
		S26	−61.9	1,911	3,973	839	6,723	2,394.2	78.5	12	5	11.94	5.3	31.0	27.3
		S27	−61.9	1,593	3,527	3,293	8,413	1,828.9	78.5	10	5	17.28	4.1	100.1	28.2
		S28	−63.7	1,297	3,526	4,088	8,911	1,600.4	78.5	6	5	13.92	9.6	57.5	27.9
	H	S29	−61.7	1,026	1,984	2,092	5,102	912.7	113.1	7	6	14.11	3.7	33.1	29.3
		S30	−60.4	526	1,335	1,789	3,650	1,011.6	113.1	5	6	9.72	5.1	17.8	28.8
	I	S31	−60.9	964	2,555	3,947	7,466	823.0	113.1	7	6	16.02	8.9	28.2	29.9
		S32	−61.0	528	1,244	982	2,754	307.9	113.1	9	6	12.72	5.4	17.2	28.0
	J	S33	−61.6	9,379	16,164	N/A	25,543	1,576.7	78.5	8	5	8.41	48.3	11.8	18.5
	K	S34	−71.0	25,988	2,894	N/A	28,882	1,380.6	1256.6	4	20	11.78	29.6	16.6	17.8
	L	S35	−77.7	23,743	28,891	N/A	52,634	935.9	1963.5	10	25	23.85	27.8	34.4	17.6
	M	S36	−79.9	8,874	12,809	N/A	21,683	1,335.2	3217.0	8	32	8.53	24.0	12.3	17.3
	N	S37	−68.5	14,250	7,479	N/A	21,729	848.1	254.5	6	9	7.83	30.6	11.3	16.8

Table 1: Summary of “Ray” database

Why RFID?

- In “Ray”, the datasets are collected by 3 gateways, referring to 3 4 × 4 antenna arrays
- Each antenna in an array is at a distance of 16cm

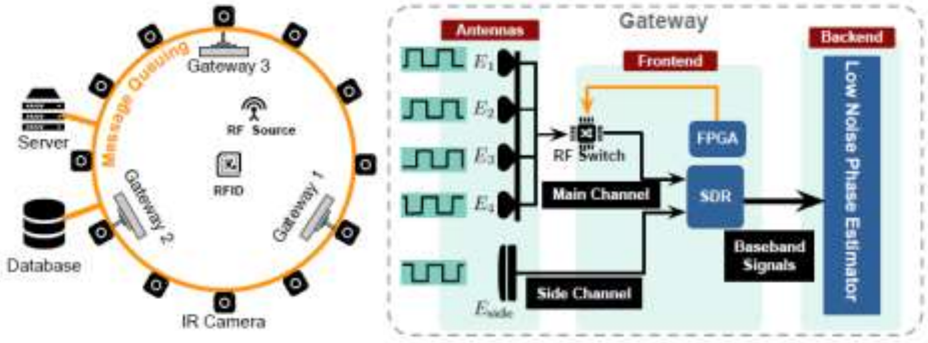

Figure 1: Deployment and architecture of the gateway

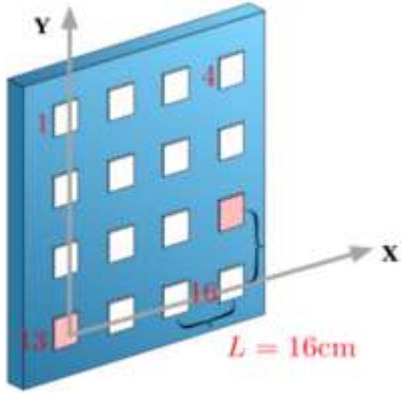

Figure 2: Arrangement of elements in a gateway

Objectives

- Method of data analysis: Deep Neural Network (DNN)
- Implementation: “ThreeBodyNet” using datasets from “Ray”
- RSSI: Strength of RF signal received by gateways from the RF identification (RFID) tag
- Phase values: Reliability of RSSI values and indication of obstructions
- Objectives:
  - Localization of the RFID tag based on the RSSI and phase values received from their positions
  - Visualization of pairs of actual and predicted coordinates of the RFID tag in a 3D Cartesian coordinate system
  - Representation of holistic performances of different prediction models in a mathematical way by cumulative distribution function (CDF)

Methodology

- The project is divided into 3 parts.
- The first part: Coordinates prediction model (project.py)
- The second part: 3D scatter plot (scatter.py)
- The third part: CDF graphs generation (cdf.py)

	project.py	scatter.py	cdf.py
Python 3.10 with Miniconda3	√	√	√
Conda 23.1.0	√	√	√
NumPy 1.23.5	√	√	√
PyTorch 1.12.1	√		
pandas 1.5.3	√		√
Dash 2.7.0		√	
Plotly 5.9.0		√	
pickle 4.0		√	
Matplotlib 3.7.1			√
openpyxl 3.0.10			√

Table 2: Technologies, frameworks, libraries, and packages used in each Python project

Coordinates prediction model

- Construct a deep neural network (ThreeBodyNet)
  - 7 layers, each layer consists of 100 neurons and applies a linear transformation to the incoming data

```python
import torch
import pandas as pd
import torch.nn as nn
from torch.utils.data import random_split, DataLoader, TensorDataset
import torch.nn.functional as F
import numpy as np

path = "Model7.pth"
# Define neural network
class Network(nn.Module):

    def __init__(self, input, output):
        super(Network, self).__init__()

        self.layer1 = nn.Linear(input, 100)
        self.layer2 = nn.Linear(100, 100)
        self.layer3 = nn.Linear(100, 100)
        self.layer4 = nn.Linear(100, 100)
        self.layer5 = nn.Linear(100, 100)
        self.layer6 = nn.Linear(100, 100)
        self.layer7 = nn.Linear(100, output)
```

Figure 3: Part of the code of project.py constructing the deep neural network

Coordinates prediction model

- Train the DNN 200 epochs
  - Using Adam optimizer with 0.0001 weight decay, 0.001 learning rate, mean squared error (MSE) loss function, training dataset (50% dataset), and validating dataset (20% dataset)
  - Each epoch computes the overall training loss and validating loss
  - Saving model with the least MSE loss function

```python
# Split train, validate and test sets
validate_set = int(len(X) * 0.2)
test_set = int(len(X) * 0.3)
train_set = len(X) - test_set - validate_set
train_set, validate_set, test_set = random_split(data, [train_set, validate_set, test_set])
# Read the data in batches and put into memory
train_loader = DataLoader(train_set, batch_size=16, shuffle=True)
validate_loader = DataLoader(validate_set, batch_size=32)
test_loader = DataLoader(test_set, batch_size=32)

# Define model
model = Network(X.size(1), Y.size(1))
model.to(torch.device("cuda:0" if torch.cuda.is_available() else "cpu"))

# Define loss function and optimizer
loss_function = nn.MSELoss()
optimizer = torch.optim.Adam(model.parameters(), lr=0.001, weight_decay=0.0001)
epochs = 200
train(model, epochs, train_loader, validate_loader, optimizer, loss_function)
print('Finish training')
```

Figure 4: Part of the code of project.py training the DNN

Coordinates prediction model

- Predict the coordinates
  - Computing test loss
  - Using testing dataset (30% dataset)

Coordinates prediction model

- Implementation of the architecture of "ThreeBodyNet"
  - Gateway 1: Red spatial spectrum
  - Gateway 2: Green spatial spectrum
  - Gateway 3: Blue spatial spectrum
- Three strategies of training "ThreeBodyNet"
  - Strategy 1: Accept 1 spatial spectrum
  - Strategy 2: Accept 2 spatial spectrums
  - Strategy 3: Accept all 3 spatial spectrums
- Parameters: RSSI and phase values
- Scenario of input: Scene A Setting 1

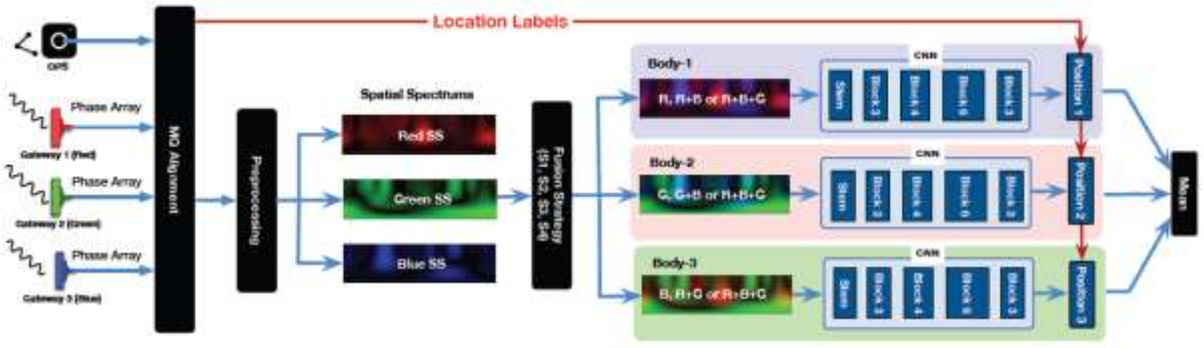

Figure 5: Architecture of "ThreeBodyNet"

Coordinates prediction model

- 7 models and prediction sets are generated in total

	Gateway 1	Gateway 2	Gateway 3
Prediction 1	✓		
Prediction 2		✓	
Prediction 3			✓
Prediction 4	✓	✓	
Prediction 5	✓		✓
Prediction 6		✓	✓
Prediction 7	✓	✓	✓

Table 3: Gateway(s) used in each prediction

3D scatter plot

- A web interface of a 3D Cartesian coordinate system
- Pairs of actual and predicted coordinates are visualized there
- Data shown:
  - Pair number
  - Actual coordinates
  - Corresponding predicted coordinates
  - Distance
  - RSSI and average RSSI
  - Phase values and average phase values

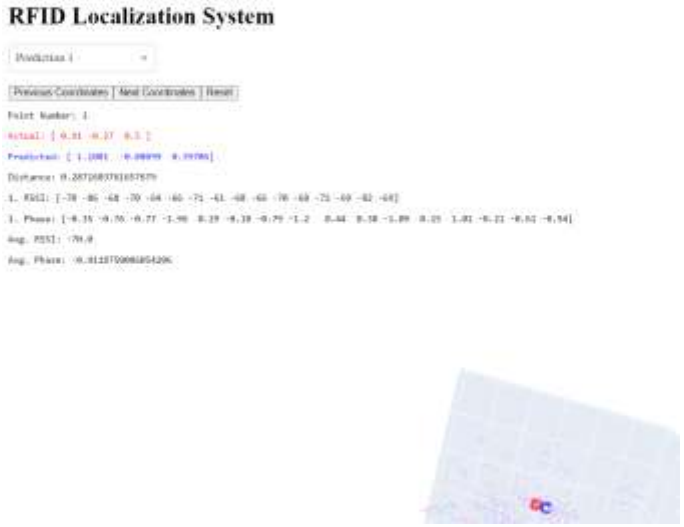

Figure 6: The 3D scatter plot webpage

CDF

- Mathematically represent the holistic performance of each prediction model to find out its extent of errors
- Greater distance of actual and predicted coordinates -> More errors
  - Parameter: Distance
- Implementation:
  - Acquisition of distances between each pair of coordinates
  - Rearrangement of the distances in ascending order

Results of “project.py”

- The results of each prediction model generated by “project.py”, the implementation of “ThreeBodyNet”

	Median of distance between actual and predicted coordinates (m)	Percentage of number of pairs of actual and predicted coordinates with distance less than 1 meter	Training loss of 200th epoch	Validation loss of 200th epoch	Overall test loss	Corresponding prediction model
Prediction 1	0.307	86.9%	0.00537	0.06350	0.09145	Model1.pth
Prediction 2	0.348	85.1%	0.06144	0.16749	0.19302	Model2.pth
Prediction 3	0.270	90.1%	0.00739	0.07595	0.07992	Model3.pth
Prediction 4	0.157	94.5%	0.02349	0.06914	0.09200	Model4.pth
Prediction 5	0.123	96.3%	0.00705	0.03506	0.08477	Model5.pth
Prediction 6	0.107	97.3%	0.00511	0.04753	0.04112	Model6.pth
Prediction 7	0.127	99.1%	0.00423	0.04459	0.03895	Model7.pth

Table 4: Summary of each prediction

Results of "project.py"

- The DNN with least test loss is adopted, generating several models and sets of prediction referring to different arrangement and number of gateways

Model1.pth	29/12/2022 0:33	PTH 檔案	216 KB
Model2.pth	29/12/2022 0:42	PTH 檔案	216 KB
Model3.pth	29/12/2022 0:51	PTH 檔案	216 KB
Model4.pth	29/12/2022 8:53	PTH 檔案	228 KB
Model5.pth	29/12/2022 9:03	PTH 檔案	228 KB
Model6.pth	29/12/2022 9:11	PTH 檔案	228 KB
Model7.pth	29/12/2022 9:22	PTH 檔案	241 KB
Prediction1.xlsx	29/12/2022 0:33	Microsoft Excel 工作...	321 KB
Prediction2.xlsx	29/12/2022 0:43	Microsoft Excel 工作...	319 KB
Prediction3.xlsx	29/12/2022 0:52	Microsoft Excel 工作...	325 KB
Prediction4.xlsx	29/12/2022 8:53	Microsoft Excel 工作...	3 KB
Prediction5.xlsx	29/12/2022 9:03	Microsoft Excel 工作...	553 KB
Prediction6.xlsx	29/12/2022 9:11	Microsoft Excel 工作...	551 KB
Prediction7.xlsx	29/12/2022 9:23	Microsoft Excel 工作...	788 KB
project.py	29/12/2022 22:26	Python File	5 KB
project.pyproj	29/12/2022 11:52	Python Project	2 KB

Figure 7: Predictions produced by respective models in "project.py"

Results of “scatter.py”

- 3D scatter plot: Starting the application results in stating the scatter plot webpage’s URL, http://127.0.0.1:8050/.
  - Entering the URL in a browser leads to the 3D scatter plot webpage.
  - Finding out the relationship between RSSI, phase value, and the distance of actual to predicted coordinates

```
C:\Users\cheuk\.conda\envs\miniconda3\python.exe
Dash is running on http://127.0.0.1:8050/

 * Serving Flask app 'scatter'
 * Debug mode: on
```

Figure 8: URL of the 3D scatter plot webpage

Results of "scatter.py"

- Findings:
  - Less error (distance between actual and predicted coordinates) is computed given lower phase values
  - Phase values are inversely proportional to accuracy

Results of “cdf.py”

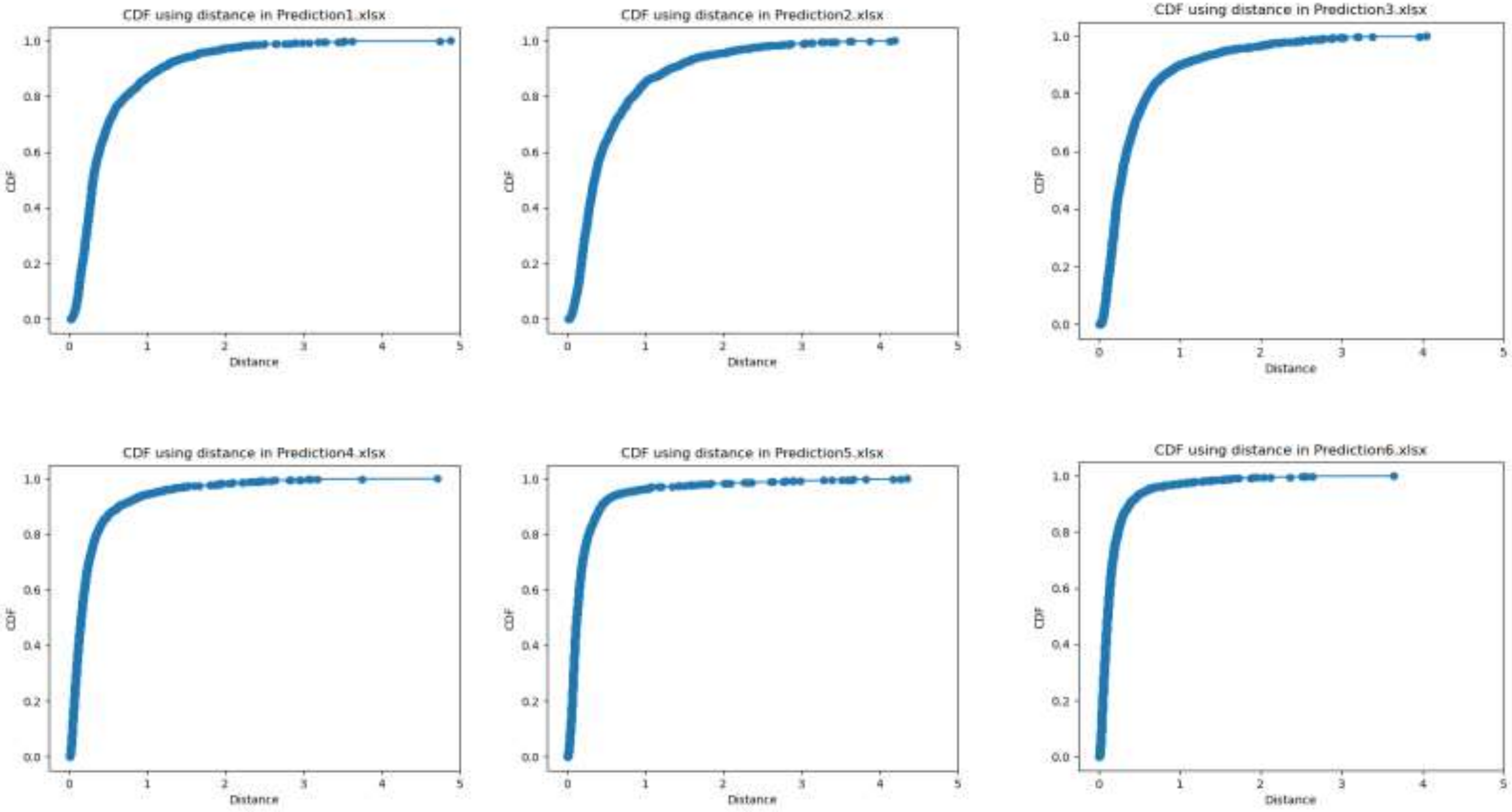

Figure 9: Cumulative Distribution Function (CDF) of distance between actual and predicted coordinates in Prediction 1-6

Results of "cdf.py"

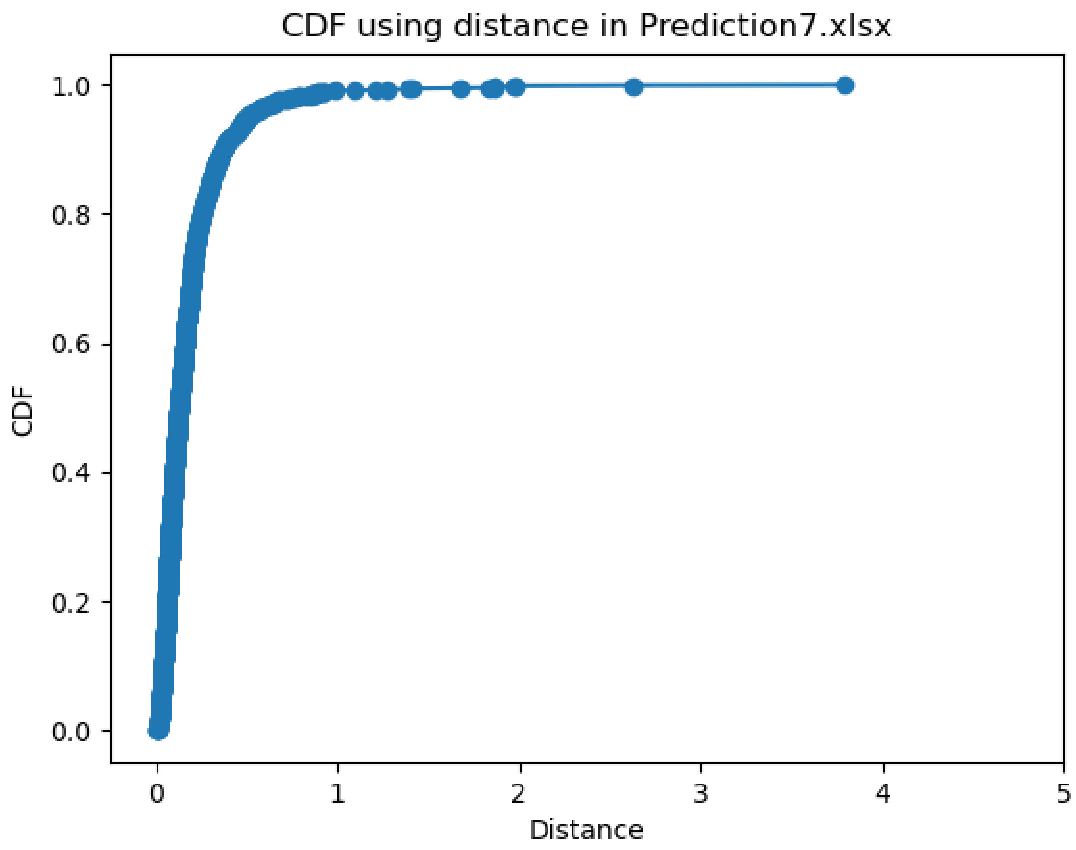

Figure 10: CDF of distance between actual and predicted coordinates in Prediction 7

Results of "cdf.py"

- Extent of inclination between the intervals 0m and 1m:
  - Prediction 1-3 (1 gateway) < Prediction 4-6 (2 gateways) < Prediction 7 (3 gateways)
- More inclined: More minor errors whereas fewer significant errors

Results from other perspective

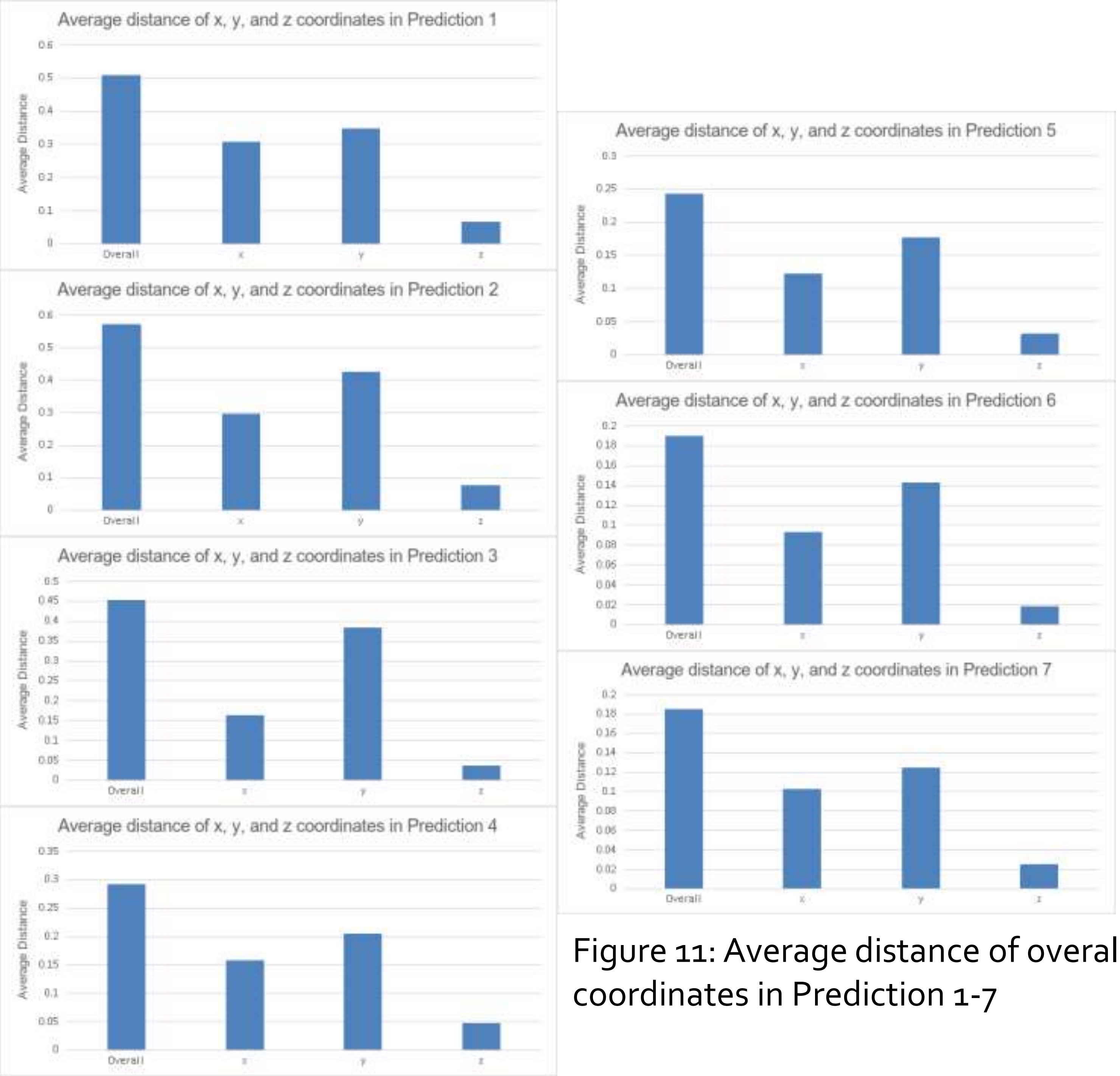

Figure 11: Average distance of overall, x, y, and z coordinates in Prediction 1-7

Results from other perspective

- Findings:
  - Error that x-coordinate contributes: 0.09m - 0.3m (slightly less than y)
  - Error that y-coordinate contributes: 0.12m - 0.42m (most)
  - Error that z-coordinate contributes: 0.02m - 0.08m (least)

Evaluations

- The performance of strategy 2 and strategy 3 exceed strategy 1 by 52.7% and 63.7% respectively
- Coverage
  - Strategy 1: 100%
  - Strategy 2: 62%
  - Strategy 3: 20%
- The rigid requirement of strategy 3 makes it hardly practical. Therefore, is prevailed to be the result of “ThreeBodyNet”

Evaluations

- Spectra generation algorithms
- Bartlett algorithm
  - Testing the assumption of equal variances of RSSI across different scenarios that the data are collected
- MVDR algorithm
  - Increasing the signal-to-noise ratio (SNR) for sets of RSSI and phase values
- Both exhibit smaller errors in x and y coordinates of within 0.1m

Evaluations

- Phase estimation algorithms in the combination of low noise phase estimator (LNPE), Kalman Filter (KF), and cyclic prefix and cyclic suffix (CC)
- LNPE
  - Estimating the relative timing and position of the signal with respect to the reference signal
- KF
  - Predicting the state of the system and updates based on the previous measurements statically with decrement of noises and other uncertainties
- CC
  - Mitigating inter-symbol interference (ISI) due to multipath propagation by cyclic extension transmitted along with the signal

Conclusion

- "Ray" database + "ThreeBodyNet" DNN = Indoor Localization Prediction
- The prediction models computed can be utilized in practical scenarios (Han, 2012)
  - Finding a particular book out of dozens of books placed on a large shelf in a library by RFID for localization prediction

References

- "Deep Learning on Indoor Localization: A Million-Scale Database with Millimeter-Level Labels," *ACM MobiCom 2022*, no. 1007, 2022.
- J. Han, Y. Zhao, Y.S. Cheng, "Improving Accuracy for 3D RFID Localization," *International Journal of Distributed Sensor Networks*, vol. 2012, no. 865184, February 2012.